Playing Abstract games with Hidden States (Spatial and Non-Spatial).

Gregory Calbert, Hing-Wah Kwok
Peter Smet, Jason Scholz,
Michael Webb VE Group, C2D, DSTO.

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **01 OCT 2003**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **Playing Abstract games with Hidden States (Spatial and Non-Spatial)**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Defence Science And Technology Organisation**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM001929. Proceedings, Held in Sydney, Australia on July 8-10, 2003., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**20**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Outline

- Our domain of research.
- The mathematics of strategically complex game playing
  - Move evaluation,
  - Min/max depth search,
  - Temporal difference learning.
- Application to our network checkers game.
- The HARD PROBLEM: hidden spatial states.
  - Information theoretic advisors,
  - Combine with TD(0)
- Open research questions.

Our domain of research

- Broadly described as medium resolution war-gaming.
  - Maneuver of forces, hidden forces, ambiguously defined end-states.
- Use machine learning techniques to develop strategies.
- Also use machine learning to capture and *generalize* expertise of human players
- Monte Carlo simulation to develop risk analysis.
- Currently looking at chess/checkers variants
  - Networked agents, Hidden agents.

DEFENCE SCIENCE & TECHNOLOGY

The Mathematics of Strategically Complex Gaming

- Often classical game theory won't work, notably because of the "tyranny of dimension."
  - Too many states and strategic paths over time.
- Solving tabular stochastic dynamic games is impossible.
- Checkers (a game of complexity in the lower end)
  - Has maneuver, materiel diversity, complex strategy
- $\approx 50^8 \approx 10^{14}$ possible strategic paths over the course of a game.
- Move to the theatre chess game (operational war-game).
- Gaming can still be viewed as approximating Bellman's state-action equation (for the optimal sequence of actions at each state and time $s_t, s_{t+1}, \cdots, s_T$) with various methods.

Evaluation

- Bellman's equation for optimal policy π^* satisfies

$$Q(s_t, a_t) = E^{\pi^*}\left\{ R(s_t, a_t) + \max_{a'} Q(s_{t+1}, a_{t+1}) \right\}$$

- The function $Q(s,a)$ is called the action-value function, specifying the *total future* reward of taking action a, from state s. Games like chess/checkers no intermediate, only terminal reward.

- Gaming playing – approximation to the action-value function by a function of a linear sum of weighted features.

$$Q \approx \tilde{Q} = f\left(\sum w_i \phi_i(s)\right)$$

- The weights w_i are called the "*advisors*" to "feature" $\phi_i(s)$ of state s (force balance).

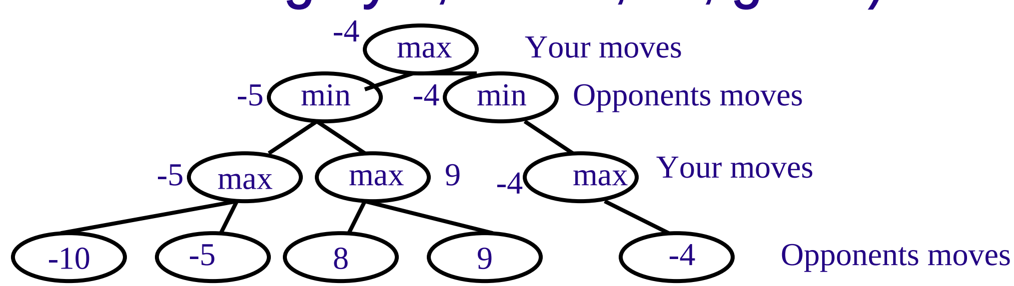
Min-Max depth search.

- Essentially you "project into the future" to a limited horizon. The optimal action is chosen to be

$$\arg\max_a \left\{\min\left(\max\left(\cdots\left(\min \tilde{Q}(s'''', a'''')\right)\right)\right)\right\}$$

- Best way to explain this is tree search, with the value at the leaves "backed up" to the root node through mini-max search.

- Limited by the branching factor of the move (checkers roughly 8, chess, 36, go 80).



Temporal difference learning.

- Remembering Bellman's equation, at the optimal policy with no intermediate rewards the difference,

$$E^{\pi^*}(Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \mid s_t, a_t) = 0$$

- In temporal difference learning you learn by "shifting" the past approximation $\tilde{Q}(s_t, a_t)$ towards the future approximation $Q(s_{t+1}, a_{t+1})$

- This incrementally minimises the "temporal difference" as is required by Bellman.

- Done by setting (if you have state numbers)

$$\tilde{Q}_{new}(s_t, a_t) = \alpha\tilde{Q}_{old}(s_{t+1}, a_{t+1}) + (1-\alpha)\tilde{Q}_{old}(s_t, a_t).$$

- The learning rate α must satisfy some simple stochastic convergence conditions.

Temporal Difference in Game Playing.

- Too many states to solve, so do TD learning on the advisors w_i so we can generalize to new novel states.
- Here we use gradient descent, incrementing the vector of advisors by

$$\Delta\vec{w} = \alpha\left(\tilde{Q}(s_{t+1}, a_{t+1}) - \tilde{Q}(s_t, a_t)\right)\nabla_{\vec{w}}\tilde{Q}(s_t, a_t).$$

- Changes in advisor weights can be done on-line (as the game is played) or off-line (at the end of each game).
- Here $\tilde{Q}(s_t, a_t) = \Pr(\text{winning game} \mid s_t, a_t)$
- Equivalently get a terminal reward of 1 if win, 0 of loss.

Function Approximation

- We approximate this probability by

$$\tilde{Q}(s,a) = 1/\left(1 + \exp\left(-\sum w_i \phi(s)\right)\right)$$

- Terminal state

$$\tilde{Q}(s_{T,} \bullet) = \begin{cases} 1 \text{ for a win,} \\ 1/2 \text{ for a draw,} \\ 0 \text{ for a loss.} \end{cases}$$

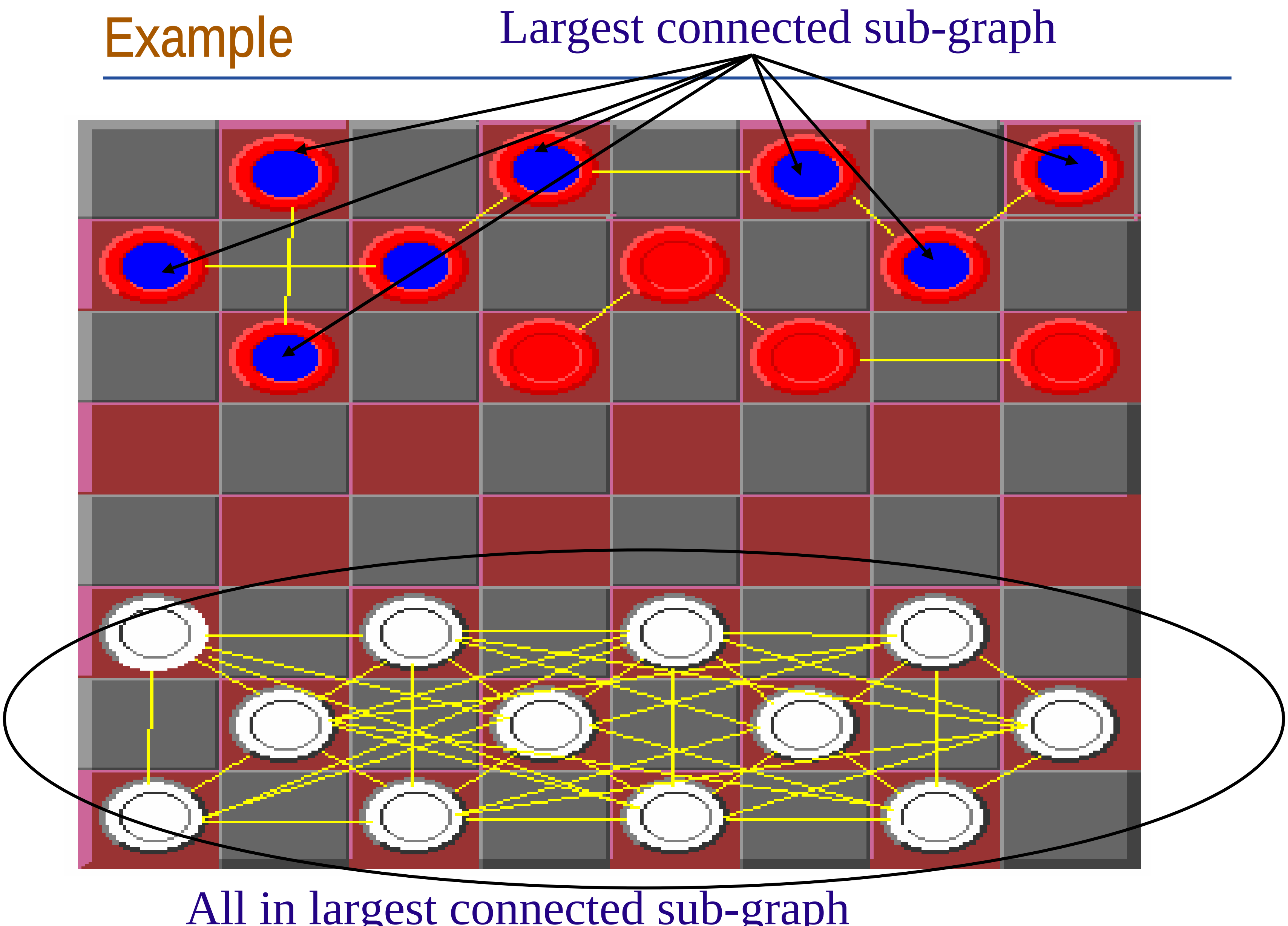
- The advisors usually reflect importance of balance in pieces, mobility, ……….. Other features of the game.
- $w_1 (N_1 - N_2) + \text{other terms}$
- N_i number of pieces of side i

An example in an imperfect information game

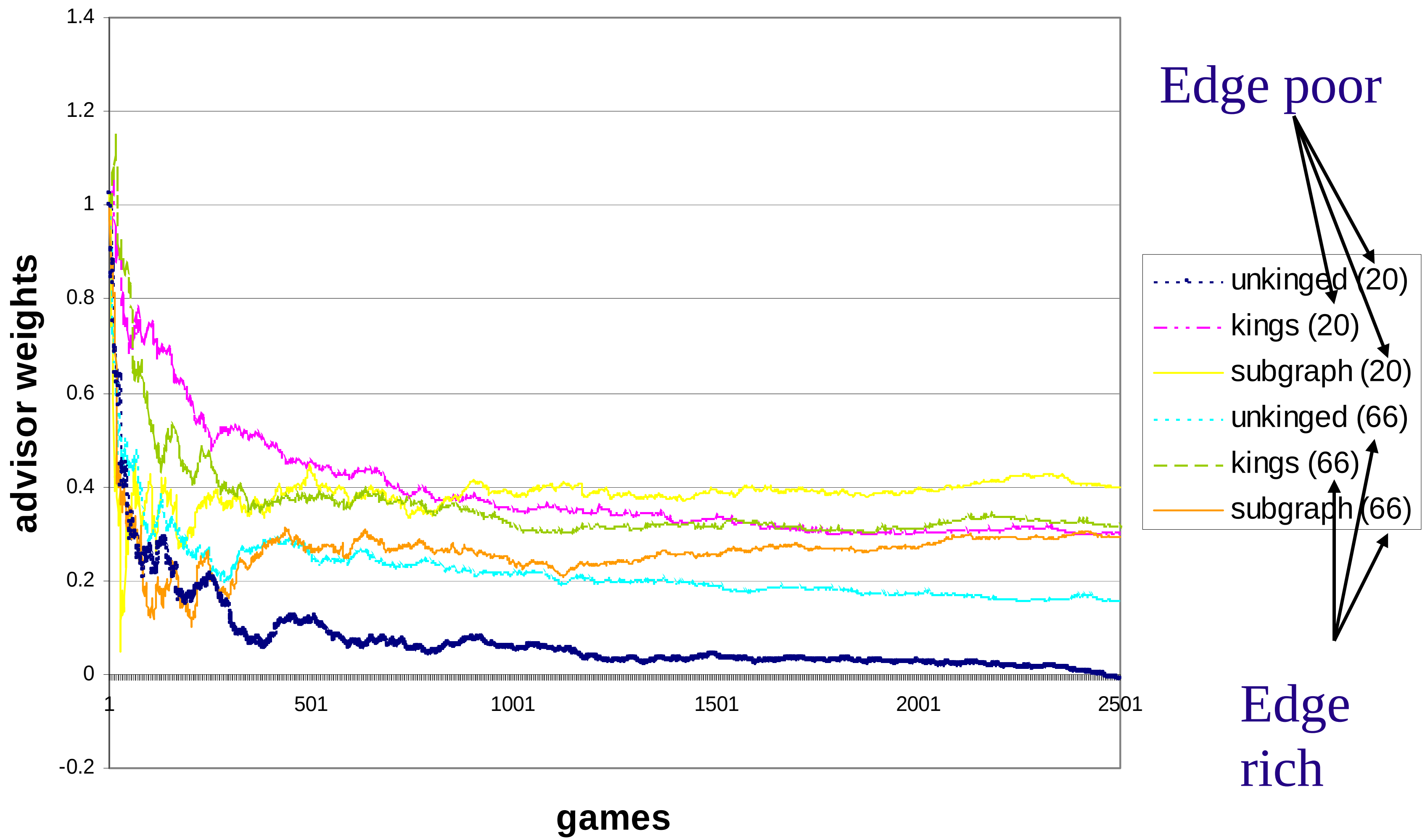
- Network checkers
  - Network vulnerability through dynamic games.
  - Pieces connected by a network of varying topology.
  - Only pieces in the *largest connected sub-graph* exhibit mobility, isolated pieces don’t.
  - Each side aware of materiel and the largest sub-graph size.
  - Network details hidden (distribution of degree etc.).

DEFENCE
SCIENCE & TECHNOLOGY

Example



Results of Learning Advisor Weights



Hidden Spatial States

- Begun research onto hidden spatial states.
- Difficult for the following reasons
  - Vastly increased branching factor of possible states. For example, suppose we have one invisible piece. Positive prob. Of being in j squares

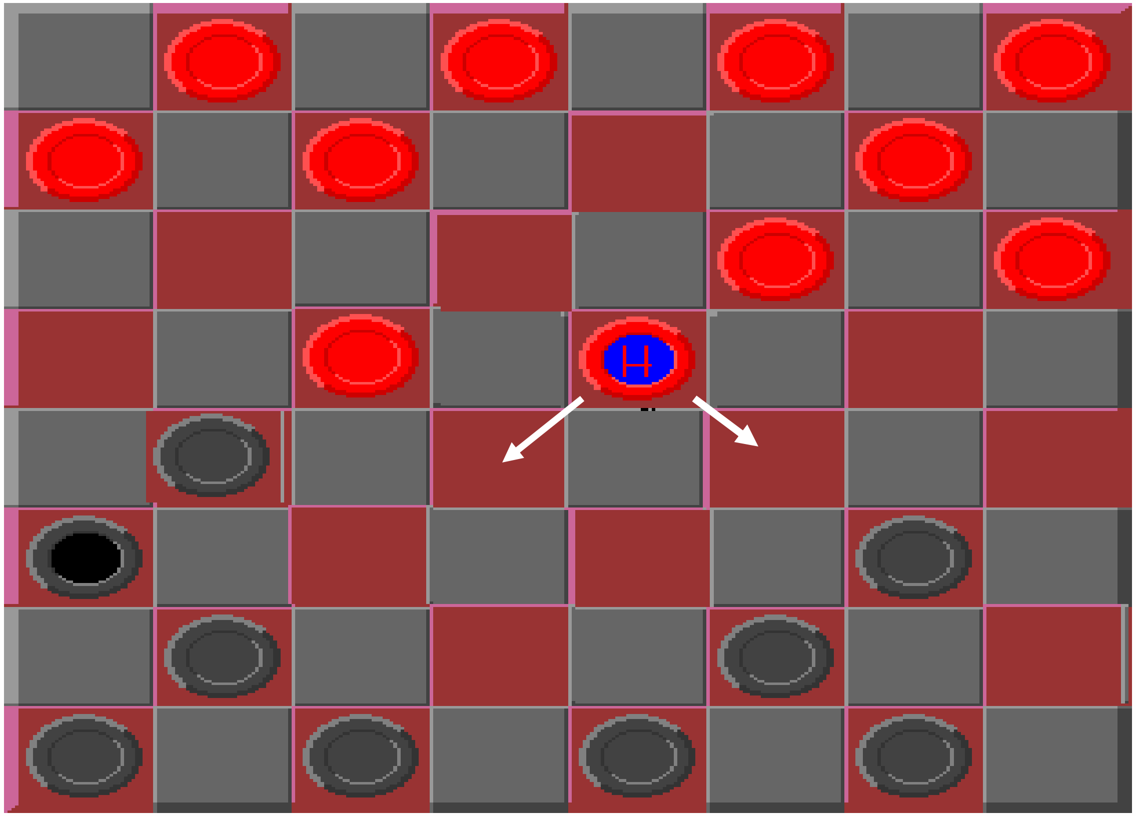
    $$New\,branching\;\;factor = Old\;branching\;\;factor * j$$

  - Have to construct a distribution of opponent's probable states. $\Pr(\text{opponent}\;\;\text{state is } s) > 0$
  - Pruning of the estimated states risky (opponent could exploit this).

Our approach

- Start with a small number of invisible pieces.
- Use theorem of total probability and conditioning on events to develop a Markov chain for the probability of hidden pieces in some state. Generate $\Pr(\text{board}_1), \cdots \Pr(\text{board}_n)$
- Really a non-Markov problem (opponent's strategies will be history dependent).
- Know when pieces are taken – including hidden piece.
- If you run into a hidden piece you loose a turn (but gain information on the hidden pieces location).
- If you try to take a hidden piece and its not there you also loose a turn, but gain information on location.

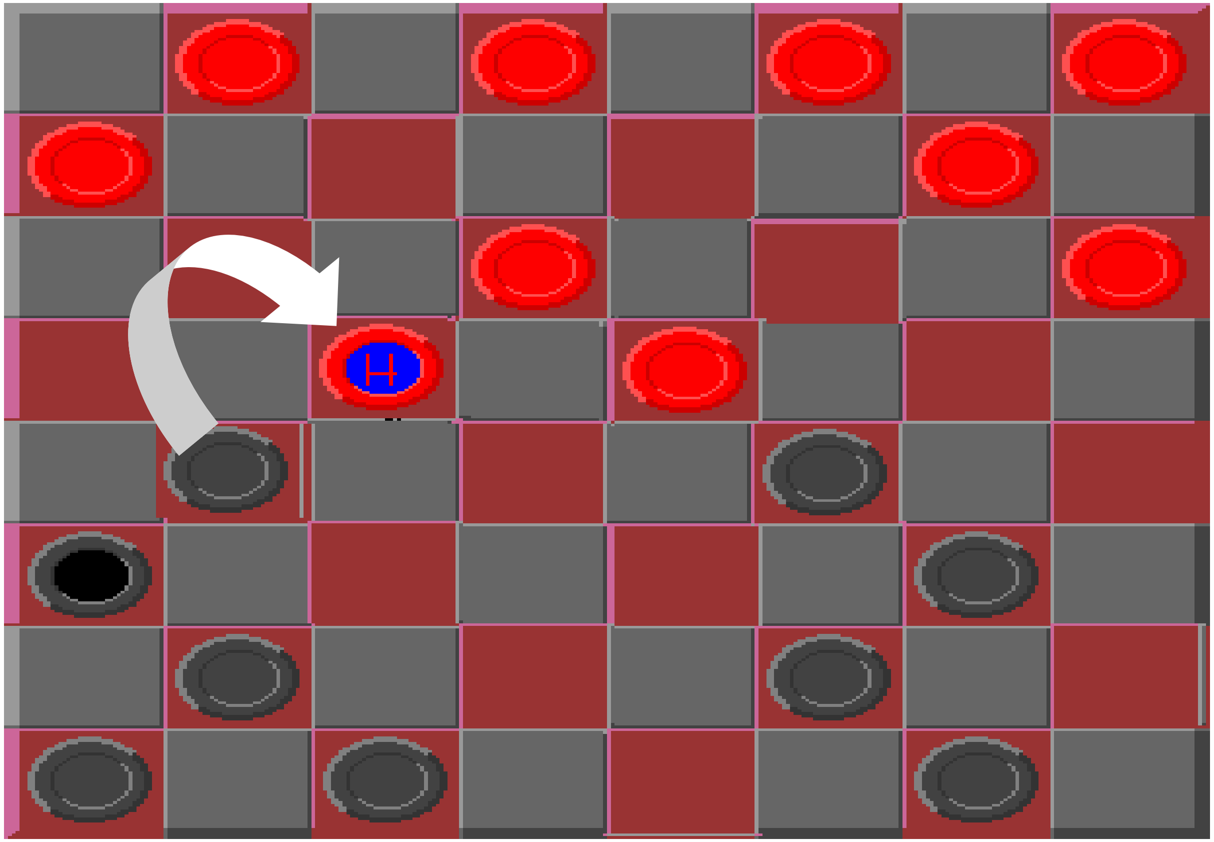
Estimation example: null move by opponent.



$$\Pr_{new}(\text{hidden at }(3,3)) = 1/2 \Pr_{old}(\text{hidden at }(4,4))$$

$$\Pr_{new}(\text{hidden at }(3,5)) = 1/2 \Pr_{old}(\text{hidden at }(4,4))$$

Estimation example: reconnaissance move by self.



$$\Pr_{\text{new}}(\text{hidden at } \mathbf{x}) =$$

$$\Pr_{\text{old}}(\text{hidden at } \mathbf{x} \mid \text{not at } (4,2))$$

Information theoretic advisor

- Opponent's movement of hidden pieces increases uncertainty of state.
- Reconnaissance moves decrease uncertainty.
- Entropy the best way to model this.
- If opponent's states have probability $p_1, p_2, \cdots; p_n$
- Then the entropy is $H = -\sum p_i \log_2 p_i$.
- We incorporate the value of reconnaissance moves through a term in the evaluation function that takes into account the entropy.

Evaluation of Moves

- Have to calculate the expectation over the possible board states. Consider moves that are legal for a particular opposition board state (with probability >0) or a reconnaissance move.
- Reconnaissance move- find out where the opponent is or isn't.
- Intend to use temporal difference learning to find the advisor weights.
- Depth of search nearly impossible, since have to carry on the same estimation/evaluation cycle.

DEFENCE
SCIENCE & TECHNOLOGY

Current research questions

- If you don’t see an opponent move
  - Was it a reconnaissance move made or a hidden move?
  - Have to learn this through Bayesian methods.
- Want to look at entropy balance
  - We`therefore have to estimate our opponents estimate of our state probabilities.
- Pruning
  - What happens when we prune boards with extremely low probability?
- Are there fast an frugal heuristics to generate strategies equal or better than the computationally expensive way?

Questions?